Implementation of Relay Agent Encapsulation for DHCPv4

Hui Deng, Lu Huang, Zhen Cao, Chentao Wang, Lubiao Cheng

China Mobile

Nov.11.2010

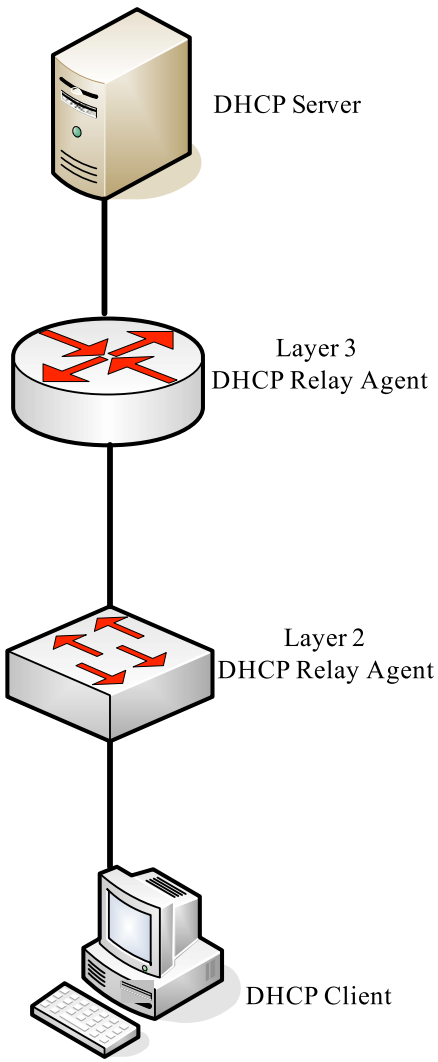
Contents

- Background
- Experimental environment
- Function we implement
- Experimental result
- Next step work

Background (1)

- Our network scenario
  - When there are two or more relay agents between client and DHCP server, Under mechanism defined in RFC3046, only one relay agent can insert relay agent information.
  - But we need to permit two or more relay agents to convey theire own information (e.g. interface ID) to DHCP server. (see draft-huang-dhc-relay-ps-00)

To solve the problem, many discussion was made. Finally Ted lead to define a mechanism similar to IPv6 in “draft-ietf-dhc-dhcpv4-relay-encapsulation-00”

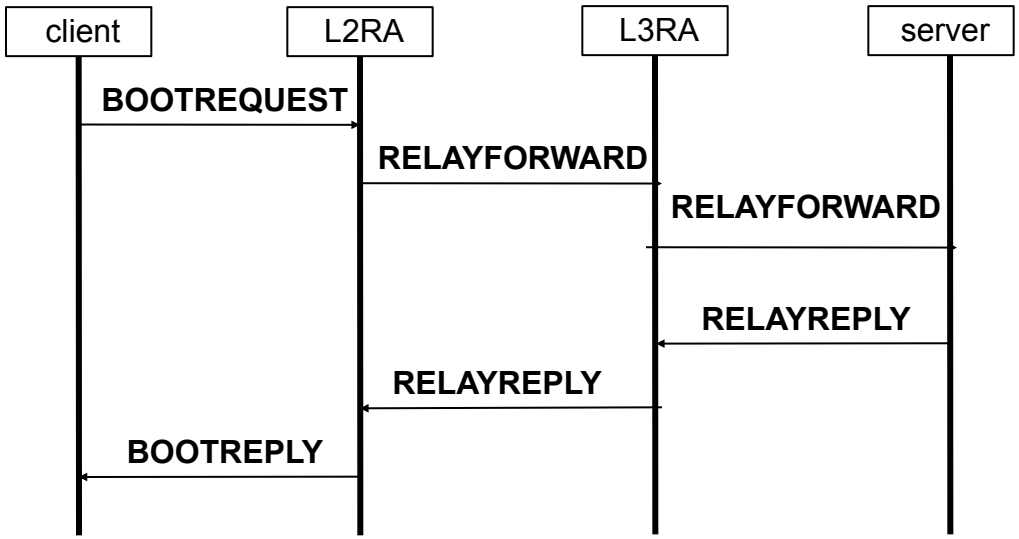


Background (2)

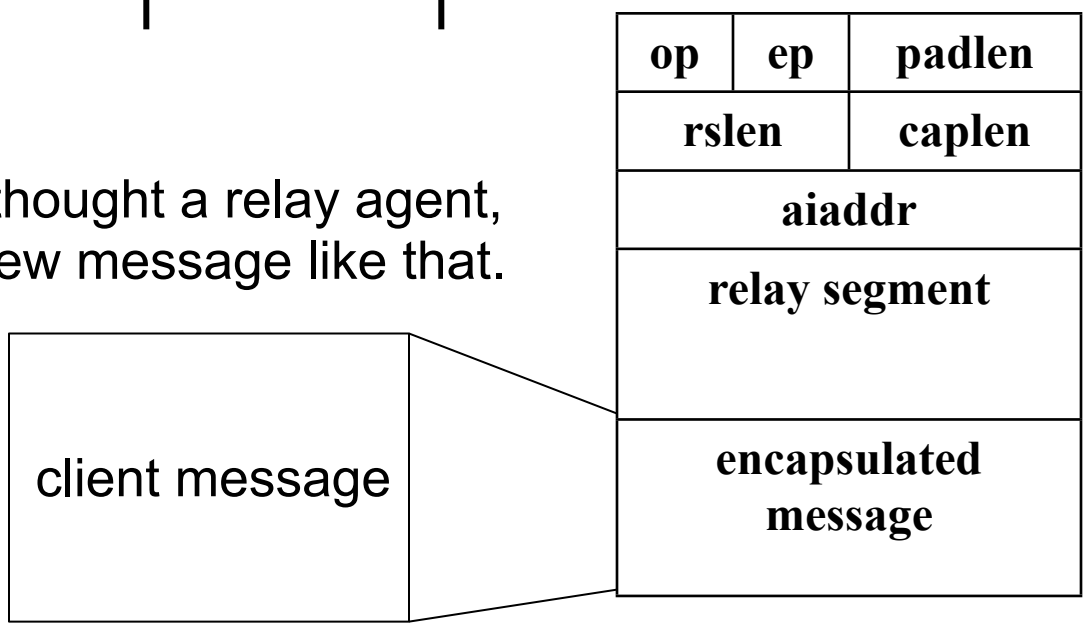
- We do our experiment according to draft-ietf-dhc-dhcpv4-relay-encapsulation-00

  The draft describes: the DHCP or BOOTP message from client to server should be encapsulated the fixed-length header and relay segment into a new message form by L2RA and L3RA. Using the encapsulation, server can know the topology of client and distribute IP address, make the forwarding decision exactly.

Background (3)



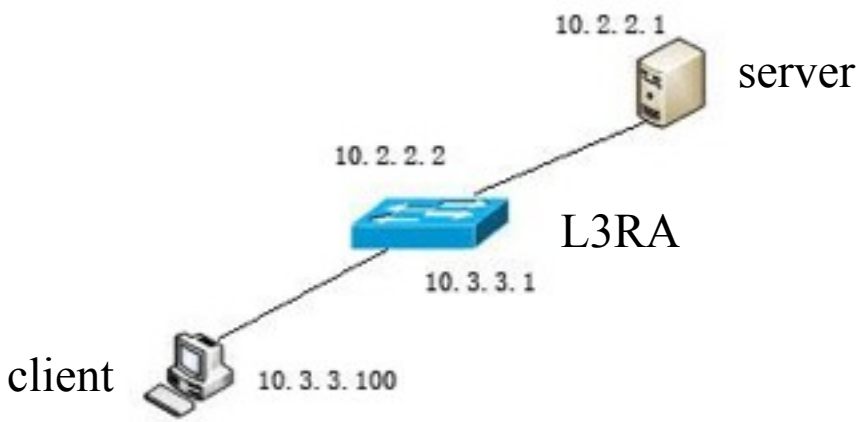
When client message go thought a relay agent, RA encapsulate it into a new message like that.



Experimental environment

- We build three hosts in VMware Workstation. One is client, One is server, the third one connecting the other two hosts with two networks is L3RA.

  Client←→L3RA←→Server



- We modify and run isc dhcp-4.2.0 version in three hosts for our experiment.

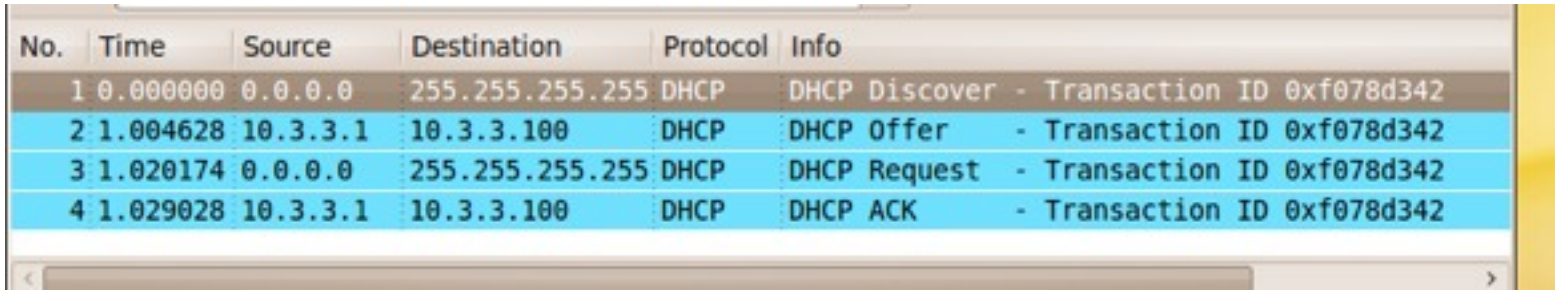
Functions we implement

- L3RA
  - Encapsulate message into a new message form.
  - Add some sub-options into relay segment, such as Circuit ID, Link selection.
  - Decapsulate RELAYREPLY message.
- Server
  - Decapsulate RELAYFORWARD message.
  - Construct RELAYREPLY message and send to relay agent.

- In order to show new packet form in wireshark software, we modify wireshark-1.4.1 code (packet-bootp.c) as this specification.

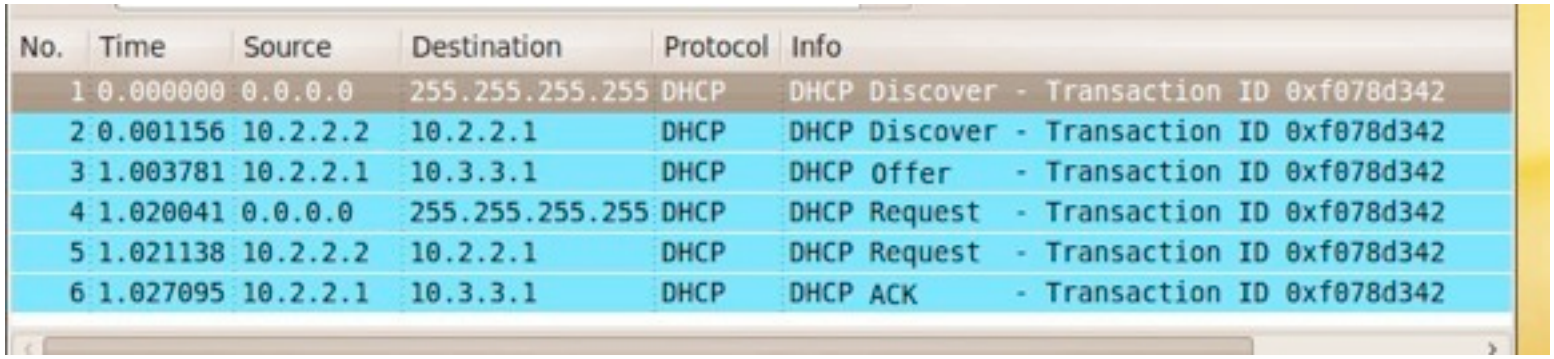
Experimental result (1)

client



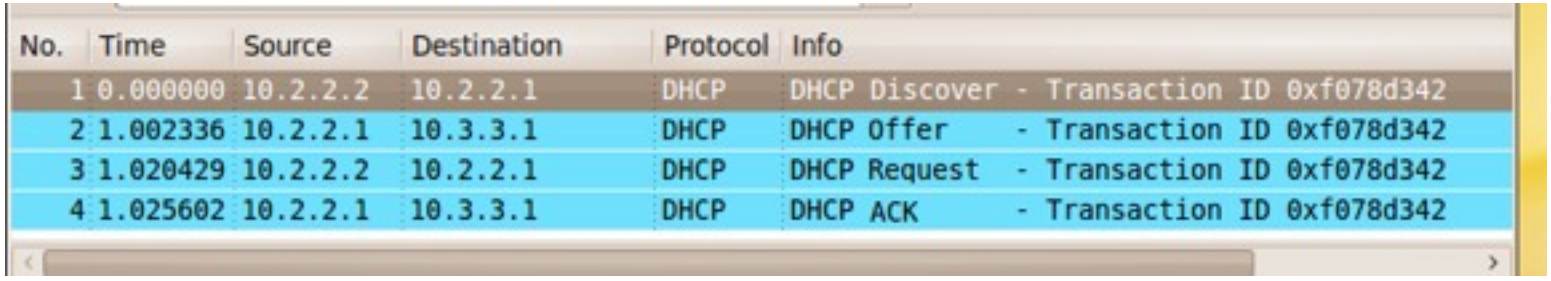
No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xf078d342
2	1.004628	10.3.3.1	10.3.3.100	DHCP	DHCP Offer - Transaction ID 0xf078d342
3	1.020174	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xf078d342
4	1.029028	10.3.3.1	10.3.3.100	DHCP	DHCP ACK - Transaction ID 0xf078d342

relay



No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xf078d342
2	0.001156	10.2.2.2	10.2.2.1	DHCP	DHCP Discover - Transaction ID 0xf078d342
3	1.003781	10.2.2.1	10.3.3.1	DHCP	DHCP Offer - Transaction ID 0xf078d342
4	1.020041	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xf078d342
5	1.021138	10.2.2.2	10.2.2.1	DHCP	DHCP Request - Transaction ID 0xf078d342
6	1.027095	10.2.2.1	10.3.3.1	DHCP	DHCP ACK - Transaction ID 0xf078d342

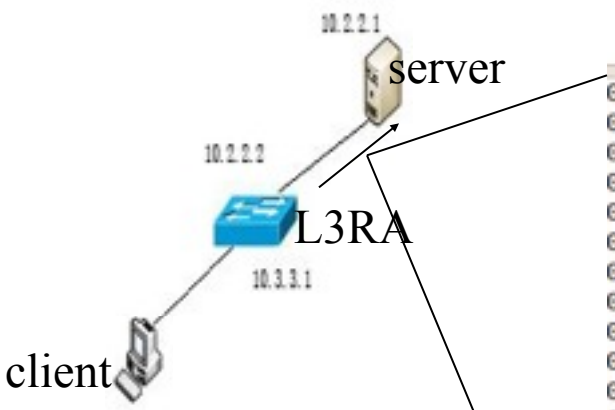
server



No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.2.2.2	10.2.2.1	DHCP	DHCP Discover - Transaction ID 0xf078d342
2	1.002336	10.2.2.1	10.3.3.1	DHCP	DHCP Offer - Transaction ID 0xf078d342
3	1.020429	10.2.2.2	10.2.2.1	DHCP	DHCP Request - Transaction ID 0xf078d342
4	1.025602	10.2.2.1	10.3.3.1	DHCP	DHCP ACK - Transaction ID 0xf078d342

Experimental result (2)

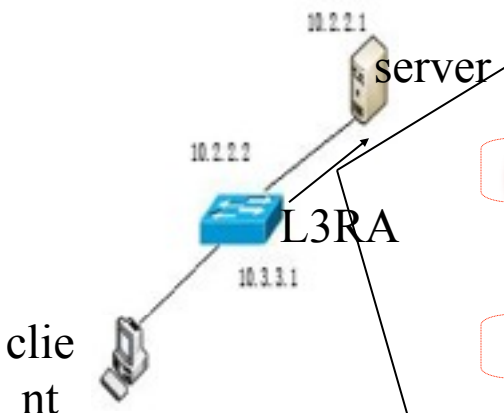
L3RA constructs the RELAYFORWARD message by adding fixed-length header and relay segment.



Fixed-length header and relay segment

```
Bootstrap Protocol
    Message type: Relay Forwrad (3)
    Does end option prior to encapsulation ? : True
    The length of padding option: 0
    The length of relay segment: 12
0000  00 04 00 01 00 06 00 0c  29 09 a5 10 00 00 08 00   ........ )....... 
0010  45 00 01 60 00 00 40 00  40 11 21 87 0a 02 02 02   E..`..@. @.!.....
0020  0a 02 02 01 00 43 00 43  01 4c d8 2e 03 01 00 00   .....C.C .L......
0030  0c 00 2c 01 0a 03 03 01  01 04 65 74 68 32 05 04   ..,..... ..eth2..
0040  0a 03 03 01 01 01 06 01  f0 78 d3 42 00 00 00 00   ........ .x.B....
0050  00 00 00 00 00 00 00 00  00 00 00 00 0a 03 03 01   ........ ........
0060  00 0c 29 b7 f1 f3 00 00  00 00 00 00 00 00 00 00   ..)..... ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0130  63 82 53 63 35 01 01 32  04 0a 03 03 64 0c 0c 77   c.Sc5..2 ....d..w
0140  61 6e 67 2d 64 65 73 6b  74 6f 70 37 0d 01 1c 02   ang-desk top7....
0150  03 0f 06 77 0c 2c 2f 1a  79 2a 52 06 01 04 65 74   ...w.,/. y*R...et
0160  68 32 ff 00 00 00 00 00  00 00 00 00 00 00 00 00   h2...... ........
```

Experimental result (3)




```
Frame 2: 368 bytes on wire (2944 bits), 368 bytes captured (2944 bits)
Linux cooked capture
Internet Protocol, Src: 10.2.2.2 (10.2.2.2), Dst: 10.2.2.1 (10.2.2.1)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Relay Forwrad (3)
    Does end option prior to encapsulation ? : True
    The length of padding option: 0
    The length of relay segment: 12
    The length of encapsulated message: 300
    Relay Agent IP address: 10.3.3.1 (10.3.3.1)
    relay segment: 010465746832050 40a030301
    Sub option: (t=1,l=4) Circuit id = "eth2"
        Sub-option: (1) Circuit id
        Length: 4
        Value: 65746832
    Sub option: (t=5,l=4) Link selection = 10.3.3.1
    encapsulated message: 01010601f078d342000000000000000000000000000000000
```

```
0010  43 00 01 00 00 00 40 00  40 11 21 87 0a 02 02 02   C.....@. @.!.....
0020  0a 02 02 01 00 43 00 43  01 4c d8 2e 03 01 00 00   .....C.C .L......
0030  00 2c 01 0a 03 03 01 01  04 65 74 68 32 05 04 0a   .,...... .eth2...
0040  03 03 01 01 01 06 01 f0  78 d3 42 00 00 00 00 00   ........ x.B.....
0050  00 00 00 00 00 00 00 00  00 00 00 00 0a 03 03 01   ........ ........
0060  00 0c 29 b7 f1 f3 00 00  00 00 00 00 00 00 00 00   ..)..... ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

padlen

op

ep

rslen

aiaddr

caplen

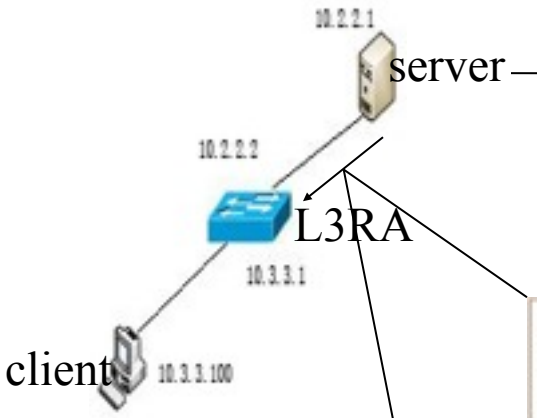
relay segment

sub-optin1

sub-option5

encapsulated message

Experimental result (4)

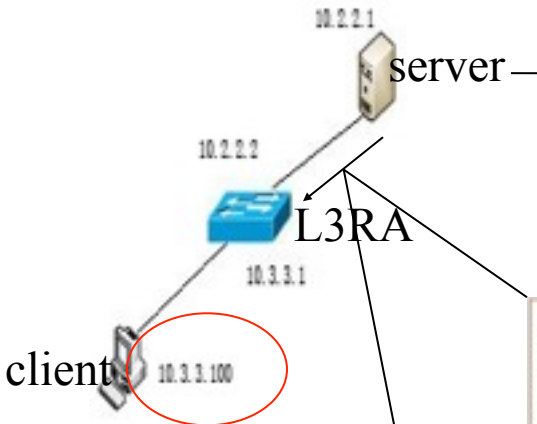



```
root@wang-desktop:~#
root@wang-desktop:~# Circuit ID = eth2
root@wang-desktop:~# Link selection = 10.3.3.1
root@wang-desktop:~#
root@wang-desktop:~#
```

```
▷ Frame 2: 376 bytes on wire (3008 bits), 376 bytes captured (3008 bits)
▷ Ethernet II, Src: Vmware_43:2c:a4 (00:0c:29:43:2c:a4), Dst: Vmware_09:a5:1(
▷ Internet Protocol, Src: 10.2.2.1 (10.2.2.1), Dst: 10.3.3.1 (10.3.3.1)
▷ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootps (67)
▽ Bootstrap Protocol
    Message type: Relay Reply (4)
    Does end option prior to encapsulation ? : True
    The length of padding option: 0
    The length of relay segment: 12
    The length of encapsulated message: 310
    Relay Agent IP address: 10.3.3.1 (10.3.3.1)
  ▽ relay segment: 0104657468320504 0a030301
    ▽ Sub option: (t=1,l=4) Circuit id = "eth2"
        Sub-option: (1) Circuit id
        Length: 4
        Value: 65746832
    ▷ Sub option: (t=5,l=4) Link selection = 10.3.3.1
    encapsulated message: 02010601f078d342000000000000000000a03036400000000..
```

```
0030  36 01 0a 03 03 01 01 04  65 74 68 32 05 04 0a 03   6....... eth2....
0040  03 01 02 01 06 01 f0 78  d3 42 00 00 00 00 00 00   .......x .B......
0050  00 00 0a 03 03 64 00 00  00 00 0a 03 03 01 00 0c   .....d.. ........
0060  29 b7 f1 f3 00 00 00 00  00 00 00 00 00 00 00 00   )....... ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Experimental result (4)




```
root@wang-desktop:~#
root@wang-desktop:~# Circuit ID = eth2
root@wang-desktop:~# Link selection = 10.3.3.1
root@wang-desktop:~#
root@wang-desktop:~#
```

```
▷ Frame 2: 376 bytes on wire (3008 bits), 376 bytes captured (3008 bits)
▷ Ethernet II, Src: Vmware_43:2c:a4 (00:0c:29:43:2c:a4), Dst: Vmware_09:a5:1(
▷ Internet Protocol, Src: 10.2.2.1 (10.2.2.1), Dst: 10.3.3.1 (10.3.3.1)
▷ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootps (67)
▽ Bootstrap Protocol
    Message type: Relay Reply (4)
    Does end option prior to encapsulation ? : True
    The length of padding option: 0
    The length of relay segment: 12
    The length of encapsulated message: 310
    Relay Agent IP address: 10.3.3.1 (10.3.3.1)
  ▽ relay segment: 0104657468320504 0a030301
    ▽ Sub option: (t=1,l=4) Circuit id = "eth2"
        Sub-option: (1) Circuit id
        Length: 4
        Value: 65746832
    ▷ Sub option: (t=5,l=4) Link selection = 10.3.3.1
    encapsulated message: 02010601f078d342000000000000000000a03036400000000..
```

```
0030  36 01 0a 03 03 01 01 04  65 74 68 32 05 04 0a 03   6.....  eth2....
0040  03 01 02 01 06 01 f0 78  d3 42 00 00 00 00 00 00   .......x .B......
0050  00 00 0a 03 03 64 00 00  00 00 0a 03 03 01 00 0c   .....d.. ........
0060  29 b7 f1 f3 00 00 00 00  00 00 00 00 00 00 00 00   )....... ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Summary & Next step work

- Summary
  - Current draft has a clear description of how to implement. Based on it, we implement a demo almost with no problem, though we don't have much experience on draft implementation.
  - The implementation proves that the new mechanism does work

- Next step work
  - Because of limitation of VMware Workstation, we did't implement L2RA yet. We will think about how to simulate a L2RA which implement the specification.
  - For redundancy, relay agent usually config two or more DHCP servers. We should consider to add mechanism to deal with scenario of two or more servers.

Thank You